This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Canceled)

2. (Currently Amended) The method of claim ~~1~~ 33 wherein said first device driver is a file system monitor.

3. (Currently Amended) The method of claim ~~1~~ 33 wherein the data is stored in a secure virtual file system, and wherein the step of performing the I/O request further comprises the step of implementing data security measures.

4. (Currently Amended) The method of claim ~~1~~ 33 wherein the data is stored in encrypted form, and wherein the step of performing the I/O request further comprises the step of decrypting the data.

5. (Currently Amended) The method of claim ~~1~~ 33 wherein the step of performing the I/O request further comprises the step of checking the data for viruses.

6. (Canceled)

7. (Canceled)

8. (Canceled)

9. (Canceled)

10. (Previously Presented) The method of claim 34 wherein the programmable security response comprises the step of destroying the data.

11. (Previously Presented) The method of claim 34 wherein the data is stored in a secure virtual file system, and wherein the step of destroying the data further comprises the step of destroying the secure virtual file system.

12. (Previously Presented) The method of claim 34 wherein the programmable security response comprises the step of terminating open applications.

13. (Previously Presented) The method of claim 34 wherein the programmable security response comprises the step of destroying said first device driver on the data storage device.

14. (Previously Presented) The method of claim 34 wherein the programmable security response comprises the step of halting the operation of the computer.

15. (Previously Presented) The method of claim 34 wherein the programmable security response comprises the step of causing the computer to enter a state requiring reboot.

16. (Canceled)

17. (Currently Amended) The method of claim ~~16~~ 35 wherein said first device driver is a file system monitor.

18. (Currently Amended) The method of claim ~~16~~ 35 further comprising a secure virtual file system for storing the data, and wherein said first device driver performs the I/O request by implementing data security measures.

19. (Currently Amended) The method of claim ~~16~~ 35 wherein the data is stored in encrypted form, and wherein said first device driver performs the I/O request by decrypting the data.

20. (Currently Amended) The method of claim ~~16~~ 35 wherein said first device driver performs the I/O request by checking the data for viruses.

21. (Canceled)

22. (Canceled)

23. (Canceled)

24. (Canceled)

25. (Previously Presented) The system of claim 36 wherein the programmable security response destroys the data.

26. (Previously Presented) The system of claim 36 further comprising a secure virtual file system for storing the data, and wherein the programmable security response destroys the data and destroys the secure virtual file system.

27. (Previously Presented) The system of claim 36 wherein the programmable security response terminates open applications.

28. (Previously Presented) The system of claim 36 wherein the programmable security response destroys said first device driver on the data storage device.

29. (Previously Presented) The system of claim 36 wherein the programmable security response halts the operation of the computer.

30. (Previously Presented) The system of claim 36 wherein the programmable security response causes the computer to enter a state requiring reboot.

31. (Canceled)

32. (Canceled)

33. (Previously Presented) A method for providing data security in a first device driver operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, the method comprising the steps of:

detecting an I/O request to said first device driver;

determining whether said first device driver has been previously called;

if said first device driver has not been previously called, detecting an initial calling module address, storing said initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver has been previously called, detecting a second calling module address, comparing said second calling module address to the initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers only if the initial calling module address matches the second calling module address;

if said first device driver is functionally uppermost in the layered plurality of device drivers, performing the I/O request in said first device driver; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, denying the I/O request in said first device driver, and allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

34. (Currently Amended) A method for providing data security in a first device driver operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, the method comprising the steps of:

detecting an I/O request to said first device driver;

determining whether said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver is functionally uppermost in the layered plurality of device drivers, performing the I/O request in said first device driver; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, denying the I/O request in said first device driver by setting a first device driver shutdown flag and initiating a re-hook process; the re-hook process comprising:

counting the number of times the re-hook process has been initiated;

checking whether the number of times has reached a predetermined maximum threshold;

if the number of times has reached a predetermined maximum threshold, initiating a programmable security response; and

if the number of times has not reached a predetermined maximum threshold, initiating reattachment of said first device driver functionally uppermost in the layered plurality of device drivers, unsetting said first device driver shutdown flag and allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

35. (Previously Presented) A system for providing data security, the system comprising a first device driver operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, wherein said first device driver:

detects an I/O request;

determines whether said first device driver has been previously called;

if said first device driver has not been previously called, detects an initial calling module address, stores said initial calling module address, and concludes that said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver has been previously called, detects a second calling module address, compares said second calling module address to the initial calling module address, and concludes that said first device driver is functionally uppermost in the layered plurality of device drivers only if the initial calling module address matches the second calling module address;

if said first device driver is functionally uppermost in the layered plurality of device drivers, performs the I/O request; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, denies the I/O request, and allows the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

36. (Previously Presented) A system for providing data security, the system comprising a first device driver operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, wherein said first device driver:

detects an I/O request;

determines whether said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver is functionally uppermost in the layered plurality of device drivers, performs the I/O request; and if said first device driver is not functionally uppermost in the layered plurality of device drivers, denies the I/O request by setting a first device driver shutdown flag and calling a re-hook system;

wherein the re-hook system comprises a counter that counts the number of times the re-hook system has been initiated to check whether the number of times has reached a predetermined maximum threshold,

if the number of times has reached a predetermined maximum threshold, the re-hook system initiates a programmable security response; and

if the number of times has not reached a predetermined maximum threshold, the re-hook system initiates reattachment of said first device driver functionally uppermost in the layered plurality of device drivers, unsets said first device driver shutdown flag and allows the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

37. (New) A machine-readable medium comprising secured data and a first device driver program for providing data security when operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, said first device driver program comprising computer-implemented instructions for:

computer-implemented instructions for detecting an I/O request to said first device driver;

computer-implemented instructions for determining whether said first device driver has been previously called;

if said first device driver has not been previously called, computer-implemented instructions for detecting an initial calling module address, storing said initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver has been previously called, computer-implemented instructions for detecting a second calling module address, comparing said second calling module address to the initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers only if the initial calling module address matches the second calling module address;

if said first device driver is functionally uppermost in the layered plurality of device drivers, computer-implemented instructions for performing the I/O request in said first device driver; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, computer-implemented instructions for denying the I/O request in said first device driver, and allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

38. (New) A computer-implemented first device driver for providing data security when operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, said first device driver comprising:

means for detecting an I/O request to said first device driver;

means for determining whether said first device driver has been previously called;

if said first device driver has not been previously called, means for detecting an initial calling module address, storing said initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver has been previously called, means for detecting a second calling module address, comparing said second calling module address to the initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of

device drivers only if the initial calling module address matches the second calling module address;

if said first device driver is functionally uppermost in the layered plurality of device drivers, means for performing the I/O request in said first device driver; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, means for denying the I/O request in said first device driver, and allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.